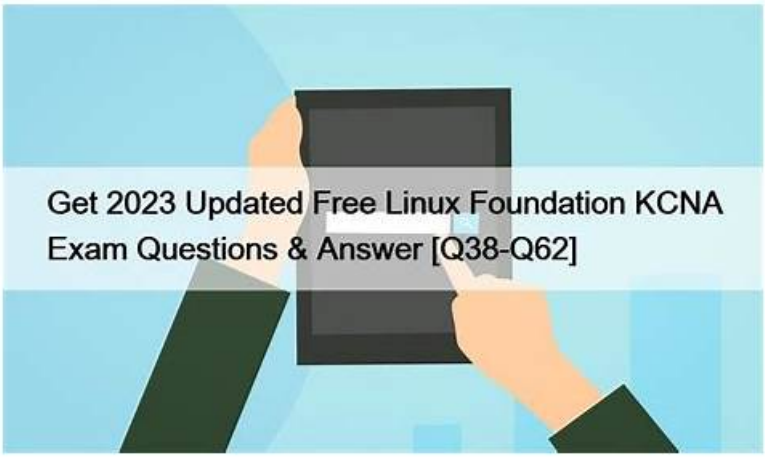
Free PDF Quiz 2026 Linux Foundation KCNA Useful Top Dumps



P.S. Free & New KCNA dumps are available on Google Drive shared by Getcertkey: https://drive.google.com/open?id=1ssp30dXPMOekfB-ITV1U3gK3qiTM5cq-

Of course, the future is full of unknowns and challenges for everyone. Even so, we all hope that we can have a bright future. Pass the KCNA exam, for most people, is an ability to live the life they want, and the realization of these goals needs to be established on a good basis of having a good job. A good job requires a certain amount of competence, and the most intuitive way to measure competence is whether you get a series of the test KCNA Certification and obtain enough qualifications.

Linux Foundation is a non-profit organization that is dedicated to promoting open-source technology, collaboration, and innovation. One of the ways in which it does this is through its certification program, which is designed to recognize and validate the skills and expertise of individuals working in the field of open-source technology. As part of this program, the Linux Foundation offers the KCNA (Kubernetes and Cloud Native Associate) exam.

>> Top KCNA Dumps <<

KCNA Practice Test Pdf, Latest KCNA Test Sample

If you are nervous on your KCNA exam for you always have the problem on the time-schedule or feeling lack of confidence on the condition that you go to the real exam room. Our Software version of KCNA study materials will be your best assistant. With the advantage of simulating the real exam environment, you can get a wonderful study experience with our KCNA Exam Prep as well as gain the best pass percentage.

The KCNA exam is intended for individuals who are new to Kubernetes and cloud-native technologies or who have a basic understanding of these technologies. It is also suitable for individuals who are planning to work with Kubernetes and cloud-native technologies in their current or future roles. KCNA Exam covers a range of topics, including Kubernetes architecture, deployment, networking, security, and troubleshooting, as well as cloud-native technologies such as microservices, service meshes, and container runtimes.

Linux Foundation Kubernetes and Cloud Native Associate Sample Questions (Q77-Q82):

NEW QUESTION # 77
Consider the following Kubernetes resource configuration for a deployment:

If each pod is consistently utilizing only 50% of the requested CPU and memory, what are the potential cost-saving measures you could take without impacting performance?

- A. Increase the 'limits' values for CPU and memory to 400m and 1024Mi, respectively.
- B. Reduce the 'requests' values for CPU and memory to 50m and 128Mi, respectively.
- C. Reduce the 'replicas' value to 2 to decrease the number of pods.
- D. Leave the current configuration as it is; the application is already optimized.
- E. Utilize a vertical pod autoscaler (VPA) to dynamically adjust the resource requests and limits based on actual usage.

Answer: B,E

Explanation:
The application is overprovisioned as pods are only utilizing 50% of requested resources. Reducing the 'requests' values to reflect actual usage (50m CPU and 128Mi memory) would optimize resource allocation without affecting performance. A VPA can further refine the resource allocation dynamically based on actual usage, ensuring optimal resource utilization. Increasing 'limits' would not reduce costs and might even increase them. Reducing replicas might impact performance, while leaving the configuration unchanged would continue overprovisioning and waste resources.

NEW QUESTION # 78
Which statement about Ingress is correct?

- A. Ingress provides a simple way to track network endpoints within a cluster.
- B. Ingress is a Service type like NodePort and ClusterIP.
- C. Ingress exposes routes from outside the cluster to Services in the cluster.
- D. Ingress is a construct that allows you to specify how a Pod is allowed to communicate.

Answer: C

Explanation:
Ingress is the Kubernetes API resource for defining external HTTP/HTTPS routing into the cluster, so D is correct. An Ingress object specifies rules such as hostnames (e.g., app.example.com), URL paths (e.g., /api), and TLS configuration, mapping those routes to Kubernetes Services. This provides Layer 7 routing capabilities beyond what a basic Service offers.
Ingress is not a Service type (so B is wrong). Service types (ClusterIP, NodePort, LoadBalancer, ExternalName) are part of the Service API and operate at Layer 4. Ingress is a separate API object that depends on an Ingress Controller to actually implement routing. The controller watches Ingress resources and configures a reverse proxy/load balancer (like NGINX, HAProxy, or a cloud load balancer integration) to enforce the desired routing. Without an Ingress Controller, creating an Ingress object alone will not route traffic.
Option A describes endpoint tracking (that's closer to Endpoints/EndpointSlice). Option C describes NetworkPolicy, which controls allowed network flows between Pods/namespaces. Ingress is about exposing and routing incoming application traffic from outside the cluster to internal Services.
So the verified correct statement is D: Ingress exposes routes from outside the cluster to Services in the cluster.

NEW QUESTION # 79
Which of the following is a correct definition of a Helm chart?

- A. A Helm chart is similar to a package and contains all the resource definitions to run an application on Kubernetes.
- B. A Helm chart is a collection of YAML files that can be applied on Kubernetes by using the kubectl tool.
- C. A Helm chart is a collection of YAML files bundled in a tar.gz file and can be applied without decompressing it.
- D. A Helm chart is a collection of JSON files and contains all the resource definitions to run an application on Kubernetes.

Answer: A

Explanation:
A Helm chart is best described as a package for Kubernetes applications, containing the resource definitions (as templates) and metadata needed to install and manage an application-so D is correct. Helm is a package manager for Kubernetes; the chart is the packaging format. Charts include a Chart.yaml (metadata), a values.yaml (default configuration values), and a templates/ directory containing Kubernetes manifests written as templates. When you install a chart, Helm renders those templates into concrete Kubernetes YAML manifests by substituting values, then applies them to the cluster.
Option A is misleading/incomplete. While charts are often distributed as a compressed tarball (.tgz), the defining feature is not "YAML bundled in tar.gz" but the packaging and templating model that supports install/upgrade/rollback. Option B is incorrect because Helm charts are not "collections of JSON files" by definition; Kubernetes resources can be expressed as YAML or JSON, but Helm charts overwhelmingly use templated YAML. Option C is incorrect because charts are not simply YAML applied by kubectl; Helm manages releases, tracks installed resources, and supports upgrades and rollbacks. Helm uses Kubernetes APIs under the hood, but the value of Helm is the lifecycle and packaging system, not "kubectl apply." In cloud-native application delivery, Helm helps standardize deployments across environments (dev/stage/prod) by externalizing configuration through values. It reduces copy/paste and supports reuse via dependencies and subcharts. Helm also supports versioning of application packages, allowing teams to upgrade predictably and roll back if needed-critical for production change management.
So, the correct and verified definition is D: a Helm chart is like a package containing the resource definitions needed to run an application on Kubernetes.

NEW QUESTION # 80
Which of the following options include only mandatory fields to create a Kubernetes object using a YAML file?

- A. apiVersion, template, kind, status
- B. apiVersion, template, kind, spec
- C. apiVersion, metadata, status, spec
- D. apiVersion, metadata, kind, spec

Answer: D

Explanation:
D is correct: the mandatory top-level fields for creating a Kubernetes object manifest are apiVersion, kind, metadata, and (for most objects you create) spec. These fields establish what the object is and what you want Kubernetes to do with it.
apiVersion tells Kubernetes which API group/version schema to use (e.g., apps/v1, v1). This determines valid fields and behavior.

kind identifies the resource type (e.g., Pod, Deployment, Service).
metadata contains identifying information like name, namespace, and labels/annotations used for organization, selection, and automation.
spec describes the desired state. Controllers and the kubelet reconcile actual state to match spec.
Why other choices are wrong:
status is not a mandatory input field. It's generally written by Kubernetes controllers and reflects observed state (conditions, readiness, assigned node, etc.). Users typically do not set status when creating objects.
template is not a universal top-level field. It exists inside some resources (notably Deployment.spec.template), but it's not a required top-level field across Kubernetes objects.
It's true that some resources can be created without a spec (or with minimal fields), but in the exam-style framing-"mandatory fields... using a YAML file"-the canonical expected set is exactly the four in D. This aligns with how Kubernetes documentation and examples present manifests: identify the API schema and kind, give object metadata, and declare desired state.
Therefore, apiVersion + metadata + kind + spec is the only option that includes only the mandatory fields, making D the verified correct answer.

NEW QUESTION # 81
Explain the concept of 'storage provisioners' in Kubernetes and how they are used to create and manage PersistentVolumes. List some popular storage provisioners and their associated storage types.

- A. Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Popular Storage Provisioners: kubernetes.io/gce-pd. Provision Google Cloud Persistent Disks kubernetes-io/aws-ebs: Provision Amazon Elastic Block Storage kubernetes.io/azure-disk: Provision Azure Managed Disks kubernetes.io/local-path: Use host directories as storage kubernetes. io/glusterfs: Use GlusterFS as a distributed file system kubernetes.io/cephfs: Use Ceph RBD as a block storage solution
- B. Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Popular Storage Provisioners: kubernetes.io/gce-pd: Provision Google Cloud Persistent Disks kubernetes.io/aws-ebs: Provision Amazon Elastic Block Storage kubernetes.io/azure-disk: Provision Azure Managed Disks kubernetes.io/local-path: Use host directories as storage kubernetes. io/glusterfs: Use GlusterFS as a distributed file system kubernetes.io/cephfs: Use Ceph RBD as a block storage solution
- C. Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Popular Storage Provisioners: kubernetes.io/gce-pd. Provision Google Cloud Persistent Disks kubernetes.io/aws-ebs: Provision Amazon Elastic Block Storage kubernetes.io/azure-disk: Provision Azure Managed Disks kubernetes.io/local-path: Use host directories as storage kubernetes.io/glusterfs: Use GlusterFS as a distributed file system kubernetes.io/cephfs: Use Ceph RBD as a block storage solution
- D. Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Popular Storage Provisioners: kubernetes.io/gce-pd- Provision Google Cloud Persistent Disks kubernetes-io/aws-ebs: Provision Amazon Elastic Block Storage kubernetes.io/azure-disk. Provision Azure Managed Disks kubernetes.io/local-path: Use host directories as storage kubernetes.io/glusterfs: Use GlusterFS as a distributed file system kubernetes.io/cephfs: Use Ceph RBD as a block storage solution
- E. Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Popular Storage Provisioners: kubernetes.io/gce-pd. Provision Google Cloud Persistent Disks kubernetes.io/aws-ebs: Provision Amazon Elastic Block Storage kubernetes.io/azure-disk: Provision Azure Managed Disks kubernetes.io/local-path: Use host directories as storage kubernetes.io/glusterfs: Use GlusterFS as a distributed file system kubernetes.io/cephfs: Use Ceph RBD as a block storage solution

Answer: B

Explanation:
Storage provisioners in Kubernetes are plugins that define the logic for creating and managing PersistentVolumes. They act as intermediaries between the Kubernetes cluster and underlying storage systems. Each provisioner is responsible for interacting with a specific storage system, like Google Cloud Persistent Disks, Amazon Elastic Block Storage, Azure Managed Disks, GlusterFS, Ceph RBD, or even local directories on the host machine. The list in option A provides some popular provisioners and their associated storage types.

NEW QUESTION # 82

.....

KCNA Practice Test Pdf: https://www.getcertkey.com/KCNA_braindumps.html

- New KCNA Exam Vce □ New KCNA Test Book ↗ KCNA Training Material □ Search for ☀ KCNA □☀□ on ➽ www.examcollectionpass.com □ immediately to obtain a free download □New KCNA Exam Vce
- Pass Guaranteed Linux Foundation - KCNA - Kubernetes and Cloud Native Associate –The Best Top Dumps □ Search for ✔ KCNA □✔□ and download it for free on " www.pdfvce.com " website □KCNA Latest Mock Test
- Top KCNA Dumps 100% Pass | Reliable KCNA Practice Test Pdf: Kubernetes and Cloud Native Associate □ Easily obtain ✔ KCNA □✔□ for free download through 「 www.exam4labs.com 」 □Free KCNA Download
- Pass Guaranteed Linux Foundation - KCNA - Kubernetes and Cloud Native Associate –The Best Top Dumps □ Simply search for ➤ KCNA □ for free download on 【 www.pdfvce.com 】 □New KCNA Braindumps Free
- Valid KCNA Exam Duration □ KCNA Latest Exam Question □ Certification KCNA Sample Questions □ Copy URL （ www.prep4away.com ） open and search for 「 KCNA 」 to download for free □KCNA Latest Exam Question
- Valid KCNA Exam Duration □ PDF KCNA Cram Exam □ KCNA Free Exam Dumps □ Open 「 www.pdfvce.com 」 and search for □ KCNA □ to download exam materials for free □Valid KCNA Exam Duration
- 2026 Latest KCNA – 100% Free Top Dumps | KCNA Practice Test Pdf □ Easily obtain ➽ KCNA □ for free download through 《 www.troytecdumps.com 》 □KCNA Latest Mock Test
- 2026 Latest KCNA – 100% Free Top Dumps | KCNA Practice Test Pdf □ Easily obtain free download of ☀ KCNA □☀□ by searching on ▸ www.pdfvce.com ◂ □KCNA Real Braindumps
- Top KCNA Dumps 100% Pass | Reliable KCNA Practice Test Pdf: Kubernetes and Cloud Native Associate □ Go to website [www.pdfdumps.com] open and search for ▹ KCNA ◃ to download for free □KCNA Download Fee
- Linux Foundation KCNA Exam Questions With Free Updates At 25% Discount □ Go to website ➡ www.pdfvce.com □□□ open and search for [KCNA] to download for free □KCNA Test Pass4sure
- Pass Guaranteed Linux Foundation - KCNA - Kubernetes and Cloud Native Associate –The Best Top Dumps □ Easily obtain □ KCNA □ for free download through □ www.practicevce.com □ □PDF KCNA Cram Exam
- www.stes.tyc.edu.tw, ehiveacademy.com, graphiskill.com, www.mixcloud.com, www.stes.tyc.edu.tw, asrschooloflaw.com, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes

P.S. Free & New KCNA dumps are available on Google Drive shared by Getcertkey: https://drive.google.com/open?id=1ssp30dXPMOekfB-ITV1U3gK3qiTM5cq-